

## Домашнее задание

1. Дан массив длины  $n$ , состоящий только из нулей и единиц. Предложите линейный алгоритм сортировки данного массива.
2. На прямой задано  $n$  отрезков, причем известно, что они образуют систему строго вложенных отрезков (их можно упорядочить так, чтобы каждый строго содержался в следующем). Отрезки заданы координатами концов  $[l_i, r_i]$  (и могут быть даны в неупорядоченном виде). Предложите асимптотически эффективный алгоритм (с точки зрения количества арифметических операций), который находит все точки прямой, которые покрыты ровно  $2n/3$  отрезками.
3. Рассмотрим детерминированный алгоритм поиска порядковой статистики за линейное время из параграфа 9.3 Кормена. Какая асимптотика будет у алгоритма, если делить элементы массива на группы по семь, а не по пять?
4. На вход задачи подаётся число  $n$  и массив чисел  $x_1, x_2, \dots, x_{2n+1}$ . Постройте линейный алгоритм, находящий число  $s$ , при котором достигается минимум суммы

$$\sum_{i=1}^{2n+1} |x_i - s|.$$

5. Предложите полиномиальный от длины входа алгоритм решения сравнения  $a \cdot x + b \equiv 0 \pmod{M}$  (На вход дают целые числа  $a, b, M$  в двоичной системе исчисления).
6. 1. Оцените глубину стека (рекурсивных вызовов) при работе быстрой сортировки в худшем случае.  
2. Измените алгоритм быстрой сортировки так, чтобы глубина стека в худшем случае была  $\Theta(\log n)$ .
7. Дан массив из  $n$  чисел. Нужно разбить этот массив на максимальное количество непрерывных подмассивов так, чтобы после сортировки элементов внутри каждого подмассива весь массив стал отсортированным. Предложите  $O(n \log n)$  алгоритм для решения этой задачи.
- 8\*. Запишите рекуррентное соотношение для сложности и докажите по индукции, что трудоемкость алгоритма Randomized-Qsort (в среднем) равна  $O(n \log n)$ .

функция  $RandomQsort(A[1..n])$

```

if  $n > 1$ 
     $x = A[RandomInteger(1, n)]$ 
    ( $x$  - случайный элемент из  $A$ )
     $Partition(A, x) \rightarrow B[1..k-1] \ x \ C[1..n-k]$ 
    (разбили  $A$  на массивы меньше  $x$ , и больше  $x$ )
     $RandomQsort(B[1..k-1])$ 
     $RandomQsort(C[1..n-k])$ 
    return  $B[1..k-1] \ x \ C[1..n-k]$ 
endif
```

9\*. Запишите рекуррентное соотношение для сложности и докажите по индукции, что трудоемкость (в среднем) алгоритма Randomized-Selection поиска  $k$ -й порядковой статистики равна  $O(n)$ .

функция  $RandomSelection(A[1..n], k)$

```
if  $n == 1$ 
    return  $A[1]$ 
else
     $x = A[RandomInteger(1, n)]$ 
    ( $x$  - случайный элемент из  $A$ )
     $Partition(A, x) \rightarrow B[1..l - 1] \ x \ C[1..n - l]$ 
    (разбили  $A$  на массивы меньше  $x$ , и больше  $x$ )
    if  $l == k$ 
        return  $x$ 
    if  $k < l$ 
        return  $RandomSelection(B[1..l - 1], k)$ 
    if  $k > l$ 
        return  $RandomSelection(C[1..n - l], k - l)$ 
```